Powershell - Active Directory

Raw LDAP queries

```
$searcher = New-Object System.DirectoryServices.DirectorySearcher
$searcher.SearchRoot = New-Object System.DirectoryServices.DirectoryEntry("LDAP://dc=....")
$searcher.SearchScope = "Subtree"
$searcher.Filter = "<ldap filter>"

foreach($result in $results) {
  $item = $results.Properties
}
```

Module

Install active directory module and then

```
Import-Module activedirectory
```

Get information about users

Below command retrieves information about administrator

```
Get-ADUser administrator
```

All users

```
Get-ADUser -filter "*"
```

Get informaton about computers

All computers

```
Get-ADComputers -filter "*"
```

Get information about groups for a user

Below command retrieves all groups for the administrator account

```
Get-ADPrincipalGroupMembership -Identity administrator
```

Get information abouts Access Control Lists Locally

Get object information from local computer:

```
Get-ACL
```

Get Security Descriptor for "C:\Windows"

```
Get-Acl C:\Windows
```

Get Security Descrpitor for log files starting with k in windows folder

```
Get-Acl -Path "C:\Windows\k*.log"
```

Get Security Descriptor for Registry key

```
Get-Acl -Path "HKLM:\System\CurrentControlSet\Control"
```

Get Security Descriptor for an object

```
Get-Acl -InputObject (Get-StorageSubsystem -Name S087)
```

Access control lists in an AD

Get object information from AD: (Spawns a new shell). Running comands in this new location retrieves ifnroamtion from AD (like Get-Acl)

```
Import-Module activedirectory
set-location ad:
Get-ACL ...
```

Otherwise you can use the distinguished name with the active directory drive mount point as prefix:

```
Get-Acl "AD:\CN=qw...,DC=...."
```

Everything below is run after setting location to AD drive

Get ACL information by using distinguished name

```
Get-Acl "OU=aaa,DC=somedomain,DC=tld"
Get-Acl (Get-ADOrganizationalUnit -filter "Name -eq 'aaa'")
```

Creates a table of permissions for the Organizational Unit aaa, in domain somedomain.tld

```
(Get-Acl (Get-ADOrganizationalUnit -filter "Name -eq 'aaa'")).Access | Format-Table IdentityReference, AccessControlType -AutoSize
```

Add user to group

```
Add-ADGroupMember <adgroup> <adprincipal>
Add-ADGroupMember SomeGroup SomUser
```

Active Directory Drive

When loading the AD module, it creates an AD drive.

Set your working location to the AD drive (makes commands run in the AD context instead, like dir)

```
Set-Location ad:
```

Set your working location to a specific domain in the AD drive

```
Set-Location "dc=somedomain,dc=local"
```

Search AD objects by SID

```
Get-ADObject -Filter "objectSid -eq 'S-1-5-321-...'"
```

Queriying trusted domains (i.e. not your current domain)

```
Get-ADUser -Server other.domain.tld "username"
```

Get ACLs for objects in trusted domains

```
New-PSDrive -Name AD2 -PSProvider ActiveDirectory -Server other.domain.tld -root "//RootDSE/"
Get-ACL ("AD2:\\" + (Get-ADUser -Server other.domain.tld "username").DistinguishedName)
```

Combination

Retreives all permissions on object for a user, and eveyr permission with only a SID, it performs a lookup to check what these SIDs are

```
(Get-Acl "AD:\$((Get-ADUser vlad).DistinguishedName)").Acess | Where-Object { $_.IdentityReference -like "s-1-*" } | Foreach { Get-A
```

GPO handling

Retrieves all GPOs (Use -Domain to specify for a specific domain)

```
Get-GPO -All
```

Retreives all inherited GPOs for a domain or OU

```
Get-GPInheritance -Target "OU=aaa,DC=somedomain,DC=local" -Domain somedomain.local
```

Retrieves applied settings from a GPO named Test (Use -All for all GPOs)

```
Get-GPOReport -Name Test -ReportType [XML|HTML] -Path C:\GPOReports\test.html
```

Retreives all permissions for a GPO named Test

```
Get-GPPermissions -Name Test -All
```

Retreives the resultant set of policy applied to a user or computer, or both

```
Get-GPResultantSetofPolicy -Path C:\GPOReports\sop.xml -ReportType XML [-Computer <>] [-User <>]
```